Profit-based Feature Selection using Support Vector Machines - General Framework and an Application for Customer Retention

Sebastián Maldonado[a], Álvaro Flores[b], Thomas Verbraken[c], Bart Baesens[c,d,e], Richard Weber[b]

[a] *Universidad de los Andes*
Mons.Álvaro del Portillo 12455, Las Condes, Santiago, Chile.
[b] *Department of Industrial Engineering, University of Chile, República 701, Santiago, Chile*
[c] *Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven, Naamsestraat 69, B-3000 Leuven, Belgium.*
[d] *School of Management, University of Southampton, United Kingdom*
[e] *Vlerick, Leuven-Gent Management School, Belgium*

Abstract

Churn prediction is an important application of classification models that identify those customers most likely to attrite based on their respective characteristics described by e.g. socio-demographic and behavioral variables. Since nowadays more and more of such features are captured and stored in the respective computational systems, an appropriate handling of the resulting information overload becomes a highly relevant issue when it comes to build customer retention systems based on churn prediction models. As a consequence, feature selection is an important step of the respective classifier construction process. Most feature selection techniques; however, are based on statistically inspired validation criteria, which not necessarily lead to models that optimize goals specified by the respective organization. In this paper we propose a profit-driven approach for classifier construction and simultaneous variable selection based on Support Vector Machines. Experimental results show that our models outperform conventional techniques for feature selection achieving superior performance with respect to business-

related goals.

Keywords: Data mining, Feature selection, Support vector machines, Churn prediction, Customer retention, Maximum profit.

---

1. Introduction

Classification is a very relevant task in many profit-driven applications, such as e.g. credit scoring or customer retention [3]. It has been shown, that the performance of a classifier can be improved by concentrating on the most relevant features used for classifier construction. Such variable selection has important advantages: first, a low-dimensional representation of the objects enhances the predictive power of classification models by decreasing their complexity. Having less features also leads to more parsimonious models which in turn contributes to reduce the risk of overfitting [9] caused by the *curse of dimensionality* [19, 30].

Additionally, it allows a better interpretation of the classifier, which is particularly important in business analytics. Many machine learning approaches are usually labeled as *black boxes* by practitioners, who therefore tend to be reluctant to use the respective methods [10]. A better understanding of the process that generates the data is therefore of crucial importance in business analytics for decision-making, e.g., by identifying those attributes that permit explaining customers' decisions [7].

In the past, statistically inspired techniques have been the most frequently used approaches to validate both, classifiers as well as feature selection methods. Recently, profit-based measures have been suggested for classifier validation [35]. In this paper we go one step further and adapt the idea of profit-driven metrics also to the task of feature selection by introducing several embedded methods combining the method Holdout Support Vector Machine (HOSVM) [26] with various validation measures.

To the best of our knowledge, profit-driven feature selection is a novel challenge that has not yet been covered in the data mining and machine

learning literature. Most of the work in business analytics and feature selection applies traditional, statistically grounded techniques without taking into account profit-related issues. Our experiments underline that the proposed approaches outperform alternative techniques and provide classifiers with highly relevant features, thus reducing the risk of overfitting while increasing the related profit at the same time.

The remainder of the paper is organized as follows: Section 2 describes the cost benefit analysis in the context of customer retention. Section 3 presents Support Vector Machines for classification and the feature selection techniques studied in this work. The proposed profit-based approach for feature selection and classification is presented in Section 4. Section 5 provides experimental results using real-world datasets. A summary of this paper can be found in Section 6, where we provide its main conclusions and address future developments.

2. The Cost Benefit Analysis Framework for Customer Retention

Trying to retain customers that are about to leave the company is one of the most important tasks in the service industry, mainly in the banking and telecommunications sector. This is driven by the increasing number of customers willing to change their provider, and the strong competition for attracting new ones. Therefore, there is an urgent need to develop and apply accurate models in order to identify current customers who are most likely to leave the company in a given period of time. Churn can be observed in two different ways, *voluntary*, meaning that the customer decides to terminate the contract, or *involuntary*, where the company decides to finish the contract with the customer [4]. In the present work we focus on churn as a voluntary decision.

If a company is able to identify potential churners, the next step is to develop marketing campaigns, and retention strategies focussing on this particular group, thus enhancing customer loyalty and leading to major benefits,

such as e.g.:

- Loyal engaged customers, can generate 1.7 times more revenue than other customers [18].
- A direct impact on profitability: a 5% increment in the customer retention rate may lead to a 18% reduction in operational costs [18].
- A decrease of money misspending, focusing resources on churn candidates instead of the whole customer database, reducing marketing and operational costs [15].

According to this, the churn rate is explicitly included in the following Customer Lifetime Value (CLV) formula [4]:

$$CLV = \sum_{t=1}^{\infty} \frac{m(1-c)^{t-1}}{(1+r)^{t-1}} = m\frac{(1+r)}{(r+c)} \tag{1}$$

where c is the annual churn rate and m stands for the mean of the annual profit contribution per customer. Parameter r is the annual discount rate. There are two classical approaches to determine this value. The first one is the company's Weighted Average Cost of Capital (WACC). The second one is to use the discount rate of the particular industrial sector. Given this formula, and understanding the CLV as the net present value of the profit for a customer, a decreasing churn rate will impact heavily on the company's profitability.

Churn phenomena can be modeled either with time-dependent techniques [4], or with single period future predictions. In the first category, this kind of models tries to not assume that the churn will occur in a given period, determining probabilities of churning up to a number of months, and taking into consideration time-varying covariates [4]. In the latter, we find approaches aiming to predict if a customer decides to churn in the next period, where the most common approaches are based on statistical methods, such

as logistic regression [8, 23, 29], non-parametric statistical models such as k-nearest neighbor [13], decision trees [39], and other machine learning techniques [15, 36]. A review on customer churn prediction modeling can be found in [37]. Here we use SVM classifiers on a single period. Churn rates usually are below 5% [35] for this kind of classification models, leading to the class-imbalance problem as will be seen in Section 5.

3. Feature Selection for SVM

In this section we present the foundations of SVM for binary classification and the different feature selection strategies available in the literature, providing a brief description of each method used in this work.

3.1. Binary Classification with Support Vector Machine

Among existing classification methods, Support Vector Machine provides several advantages such as adequate generalization to new objects due to the *Structural Risk Minimization* principle, absence of local minima via convex optimization, and representation that depends on only a few data points (the *support vectors*). All these features reduce the risk of overfitting in classification [34]. Additionally, the introduction of kernel functions for nonlinear classification enhances performance via flexible classifiers, in contrast to traditional techniques such as logistic regression.

Let $\mathcal{F}$ be the feature set, $\mathbf{x}_i \in \Re^{|\mathcal{F}|}$ the feature vector and $y_i \in \{-1, 1\}$ the class label of object i, $i = 1, ..., N$. $\mathcal{T} = \{(\mathbf{x}_i, y_i); i = 1, ..., N\}$ denotes the training set.

In our case, $\mathbf{x_i}$ is the feature vector describing customer i and y_i indicates his/her class label; churners (non-churners) are identified by $y_i = 1$ ($y_i = -1$).

Linear SVM constructs an optimal hyperplane $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ which tries to correctly separate one class from the other. To achieve this optimal hyperplane, SVM aims to maximize its *margin*, defined as the sum of the distances (with a given metric) between the hyperplane to the closest positive and negative training patterns. This is equivalent to minimizing the

Euclidian norm of $\mathbf{w}$ [34]. Given that a perfect separation between the two classes is not always possible, a slack variable ξ_i is introduced for each training vector $\mathbf{x_i}$, $i = 1, ..., N$ whereby C is used as a penalization parameter to control the training error [34] as shown in model (2).

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\left\|\mathbf{w}\right\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i \cdot (\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \qquad i = 1, \ldots, N, \\ & \xi_i \geq 0, \qquad\qquad i = 1, \ldots, N. \end{aligned} \tag{2}$$

The previous formulation can be extended to non-linear classifiers by using the *kernel trick*: the training samples are mapped into a higher dimensional domain $\mathscr{H}$ through the function $\phi : \mathbf{x} \to \phi(\mathbf{x}) \in \mathscr{H}$ [31]. A kernel function $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^\top \cdot \phi(\mathbf{y})$ defines an inner product in space $\mathscr{H}$, leading to the following dual formulation:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,s=1}^{N}\alpha_i\alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s) \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, N. \end{aligned} \tag{3}$$

In this work we use both, linear SVM as well as the kernel-based formulation with Gaussian kernel, which usually achieves very good results and is a common choice in the literature [26, 31]. This kernel function has the following form:

$$K(\mathbf{x}_i, \mathbf{x}_s) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_s||^2}{2\sigma^2}\right) \tag{4}$$

where $\sigma > 0$ controls the kernel width.

3.2. Related Work on Feature Selection

There are three main approaches developed for Feature Selection (for further information see [19]):

- **Filter methods**: These methods take place before applying any classification algorithm, and use statistical properties of the features aiming at filtering out the ones that contribute less information. Classical examples are χ^2 *statistic*, which measures how independent the distribution of each feature against the class labels is [33], *Information Gain* (also known as Mutual Information) which uses information entropy in order to decide how relevant a given feature is [5, 33], and the *Fisher Criterion Score (F)*, which estimates each feature's relevance independently of the others, as shown in (5):

$$F(j) = \left| \frac{\mu_j^+ - \mu_j^-}{(\sigma_j^+)^2 + (\sigma_j^-)^2} \right| \tag{5}$$

  where μ_j^+ (μ_j^-) is the mean of the j-th feature's values in the positive (negative) class and σ_j^+ (σ_j^-) is the corresponding standard deviation.

- **Wrapper methods**: These methods go through the set of features in order to score possible feature subsets regarding their predictive potential. This approach is computationally demanding because it has an exponential size on the input, but in most cases provides better results than filter methods [19, 24]. The most popular wrapper strategies are *Sequential Forward Selection (SFS)* and *Sequential Backward Elimination (SBE)*. SFS starts with an empty set of features, and then tries out the features one at a time, and includes in each iteration the most relevant one (according to a particular classification method) of the remaining set. SBS is the opposite of the first method, starting with the

entire feature set and calculates one by one their statistical significance, eliminating in each iteration the least significant one.

- **Embedded methods**: These models perform feature selection simultaneously with classifier construction, which means searching in the combined space of both hypotheses and features. Similar to wrapper methods, this approach is specific for each classification method, and therefore includes the relation of the feature dependencies with the classifier. However, embedded methods are computationally less intensive than wrapper methods [19].

These three feature selection strategies are depicted in Figure 1. Backward elimination approaches are the most common strategy for wrapper or embedded feature selection due to the before-mentioned advantages [19]. Therefore we used this strategy for feature selection.

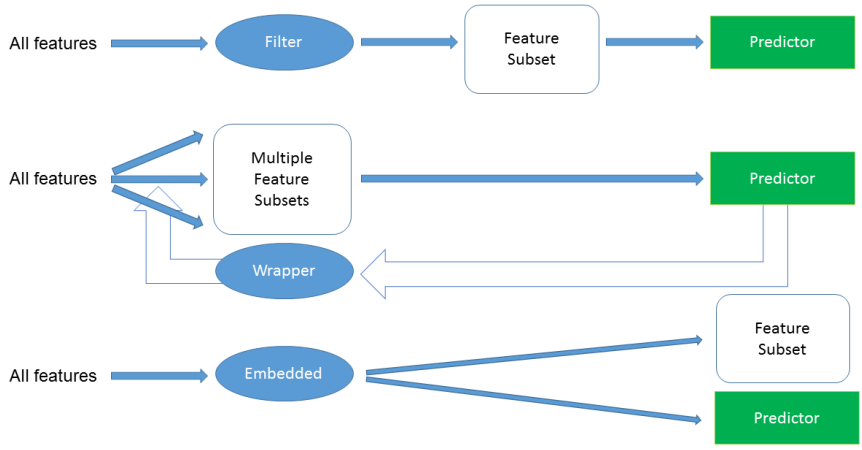

Figure 1: Feature selection strategies

One popular embedded method, which is relevant for the remainder of this paper, is known as Recursive Feature Elimination (RFE-SVM) [20]. The goal

of this approach is to find a subset of size r among $|\mathcal{F}|$ variables ($r < |\mathcal{F}|$), eliminating those features whose removal leads to the largest margin of class separation. Since the margin is inversely proportional to the Euclidean norm of the weight vector, this value can be rewritten in terms of the dual variables of SVM:

$$W^2(\boldsymbol{\alpha}) = \sum_{i,s=1}^{N} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s). \tag{6}$$

The RFE-SVM algorithm can be extended in several ways. In particular, in [26] we proposed a modification of the contribution measure (6) based on the misclassification errors instead of the margin. The backward elimination algorithm was also modified, including a holdout step: the classifier was trained on a training subset, while the number of misclassified instances was computed on a validation subset extracted from the training set, leading to the HOSVM method.

Feature selection can also be an unsupervised task [1]. Unsupervised feature selection focus on a *target concept* rather than on class labels, where observations that are close to each other in the feature space should belong to the same target concept [41, 42]. Some approaches that follow this principle using spectral graph theory are SPEC [41] and Laplacian Score [22]. In the same context, Zhang and Hancock [40] proposed a feature selection strategy based on hypergraph clustering.

4. Proposed profit-based feature selection and classification approach

We propose different embedded methods for profit-based feature selection using Support Vector Machines which are inspired by HOSVM [26]. The rationale behind our approaches is that we eliminate those features whose removal has less impact in the final solution, considering the respective costs and benefits. This will be measured using profit-based metrics, namely MPC and EMPC, as well as using the H measure, leading to the feature selection

methods called HOSVM$_{MPC}$, HOSVM$_{EMPC}$, and HOSVM$_{H}$, respectively.

To evaluate the respective models, a variety of performance measures has been proposed in the literature [35]. Section 4.2 provides a detailed description of such metrics. In Section 4.3 we present our methods for profit-based feature selection and classification. While in this paper we focus on churn prediction, the respective methods are introduced in a rather generic way to facilitate their use in different applications.

4.1. Notation and preliminaries

For a given sample $\mathbf{x}$, a classifier $\mathcal{C}$ will produce a score $s \in [0, 1]$, where by convention a high score means the corresponding customer is more likely to churn. A threshold value t is defined to provide a crisp classification of all customers based on their scores. All instances with $s \leq t$ are classified as non-churners ($y = -1$), whereas instances for which $s > t$ are classified as churners ($y = 1$).

Following the notation presented in [35], we define:

- **Prior probabilities**: π_{-1} and π_1 are the prior probabilities of a given sample to belong to class -1 or 1, respectively.
- **Probability distributions**: For a given score s, the probability density functions for non-churners and churners are $f_{-1}(s)$ and $f_1(s)$, respectively, whereas the cumulative density functions are denoted by $F_{-1}(s)$ and $F_1(s)$.
- **Cost-benefit terms**: We define b_{-1} (b_1) as the benefit of a correctly classified non-churner (churner), and c_{-1} (c_1) as the cost of a misclassified non-churner (churner). We also define $\theta = (b_1 + c_1)/(b_{-1} + c_{-1})$ as the *cost benefit ratio* to simplify notation. Both, the optimal threshold as well as the profit will depend only on this ratio of costs and benefits.

4.2. Classical and profit-based measures for model validation in churn prediction

The literature proposes various performance measures for classification models; see e.g. [12]. A frequently used measure is the AUC, which is the area under the ROC curve. A receiver operating characteristic (ROC) curve is a graphical representation of the classification performance with varying threshold t, or, in other words, a plot of the sensitivity versus one minus the specificity, i.e. $F_{-1}(t)$ as a function of $F_1(t)$:

$$\begin{aligned}
&\text{Sensitivity} = F_{-1}(t),\\
&\text{Specificity} = 1 - F_1(t),\\
&\text{AUC} = \int F_{-1}(s) f_1(s) \mathrm{d}s.
\end{aligned}$$

The area under the ROC-curve is often used to assess a classifier's performance. In simple terms, the AUC of a classification method is the probability that a randomly chosen positive observation will be ranked higher than a randomly chosen negative one [16]. A larger AUC indicates better performance. Sensitivity and specificity are useful to compute the AUC, and also to obtain the contribution metrics used in our proposal. In particular, the profit function is computed based on sensitivity and specificity.

The problem with traditional measures, such as AUC, is that they implicitly make unrealistic assumptions about misclassification costs [21, 38]. Several performance metrics have been proposed to overcome this problem. Those relevant for the present work on churn prediction will be described next.

When setting up a customer retention campaign, a fraction η of the customers with the highest propensity to churn is contacted (incurring a cost of f per person) and an incentive to stay leading to a monetary cost d is offered. Among these customers there are true would-be churners and false would-be

churners. We assume that in the latter group everyone accepts the incentive and does not churn, because they did not have the intention in the first place [38]. For the former group, on the other hand, a fraction γ accepts the offer and thus results in an earned customer lifetime value (CLV) (Equation 1), whereas the fraction $(1-\gamma)$ effectively churns despite the incentive. In the other fraction $(1-\eta)$ of customers, which is not contacted, all would-be churners churn, and all non-churners stay with the company. This process is summarized in the following formula [29]:

$$\text{Profit } = N\eta\left[(\gamma CLV + d(1-\gamma))\,\pi_{-1}\lambda - d - f\right] - A, \tag{7}$$

where η is the targeted fraction of customers, CLV is the customer lifetime value (see Equation 1), d is the cost of the incentive (offer), f is the cost of contacting the customer, and A are the fixed administrative costs. The lift coefficient λ is the fraction of churners in the targeted fraction η of customers divided by the base churn rate for all the customers π_{-1} [35]. Finally, γ is interpreted as the probability of a contacted churner accepting the incentive and thus not churning. Here CLV, A, f, and d are positive, and for coherence $CLV > d$. In this scheme it is clear that η depends on the choice of the threshold t, and this enables the company to decide upon the fraction of customers to be targeted by the retention campaign.

If we study the average profit instead of the total profit, A can be discarded because it is a fixed cost, independent of the classifier. We define the *average classification profit of a classifier for customer churn* as follows:

$$P_{\mathcal{C}}(t;\gamma,CLV,\delta,\phi) = CLV\,(\gamma(1-\delta)-\phi)\,\pi_{-1}F_{-1}(t) - CLV(\delta+\phi)\cdot\pi_1 F_1(t). \tag{8}$$

where $\delta = \frac{d}{CLV}$ and $\phi = \frac{f}{CLV}$. We also note that $b_{-1} = CLV\,(\gamma(1-\delta)-\phi)$, and $c_1 = CLV(\delta+\phi)$.

Given a training set of customers $\mathcal{T}$ with features $\mathcal{F}$ we study three different measures, described as follows:

- **The Maximum Profit Criterion for Customer Churn (MPC)**: If we assume that all the parameters in Equation (8) are known, for a given classifier $\mathcal{C}$ we will obtain a deterministic performance measure. Taking the maximum value over all possible thresholds, we have the following assessment metric [35]:

$$\text{MPC} = \max_t P_{\mathcal{C}}(t; \gamma, CLV, \delta, \phi). \tag{9}$$

  This way we obtain the fraction of the customers that should be targeted $\bar{\eta}_{\text{mpc}}$ in order to maximize the profit generated by the retention campaign, given by:

$$\bar{\eta}_{\text{mpc}} = \pi_{-1} F_{-1}(T) + \pi_1 F_1(T), \tag{10}$$

  with

$$T(\gamma) = \arg\max_t P_{\mathcal{C}}(t; \gamma, CLV, \delta, \phi). \tag{11}$$

- **The Expected Maximum Profit Measure for Customer Churn (EMPC)**: In this particular case and following [38], we model γ, the probability of a churner accepting the incentive, as a Beta distributed random variable, leading to the following formula for the *Expected Maximum Profit* for a classifier $\mathcal{C}$:

$$\text{EMPC} = \int_\gamma P_{\mathcal{C}}(T(\gamma); \gamma, CLV, \delta, \phi) \cdot h(\gamma) \mathrm{d}\gamma, \tag{12}$$

  with $T(\gamma)$ being the optimal threshold (Equation (11)) and $h(\gamma)$ the probability density function for γ. The parameters α and β related to the Beta distribution of γ were obtained from a previous work in churn

prediction [38]. Analogous to MPC, the percentage of the customers targeted in the retention campaign for this metric follows:

$$\bar{\eta}_{\text{EMPC}} = \int_{\gamma} \left[\pi_{-1} F_{-1}\left(T(\gamma)\right) + \pi_1 F_1(T\left(\gamma\right))\right] \cdot h(\gamma) \mathrm{d}\gamma, \tag{13}$$

It is important to note the influence of γ in this equation since it has a direct impact on the cost benefit ratio:

$$\theta = \frac{b_1 + c_1}{b_{-1} + c_{-1}} = \frac{\delta + \phi}{\gamma(1-\delta) - \phi} \tag{14}$$

- **The H-Measure**: Hand [21] proposed the H measure as an alternative to the AUC. The difference between H-measure and MP-measures is that H only focusses on costs. Hence, the focus is not on the expected maximum profit, but on the expected minimum loss, defining the average classification loss Q as:

$$Q_{\mathcal{C}}(t; c, b) = b \cdot \left[c\pi_{-1}\left(1 - F_{-1}(t)\right) + (1-c)\pi_1 F_1(t)\right], \tag{15}$$

with $c = \frac{c_0}{c_{-1}+c_1}$ and $b = c_{-1} + c_1$. Here, the cost benefit ratio on which the optimal threshold T depends, is $\theta = \frac{1-c}{c}$.

Calculating the value of the expected minimum loss requires assumptions on the probability density functions of both b and c. Assuming that b and c are independent, and defining $w(b, c)$ as the joint probability density function of b and c, whereas $u(c)$ and $v(b)$ are the marginal probability density functions of c and b, respectively, the explicit relationship between these densities is $w(b, c) = u(c) \cdot \ v(b)$. Hence, the expected minimum loss L is equal to:

$$\mathrm{L} = \mathrm{E}[b] \int_0^1 Q_{\mathcal{C}}(T(c); b, c) \cdot u(c) \mathrm{d}c, \tag{16}$$

with $\mathrm{E}[b] = 1$ for an appropriate choice for the unit in which b is measured. We assume that c follows a Beta distribution with parameters α and β, characterized as follows:

$$u_{\alpha,\beta}(x) = \begin{cases} \frac{x^{\alpha-1}\cdot(1-x)^{\beta-1}}{B(1,\alpha,\beta)} & \text{if } x \in [0,1], \\ 0 & \text{else,} \end{cases} \tag{17}$$

with $\alpha, \beta > 1$, and:

$$B(x,\alpha,\beta) = \int_0^x t^{\alpha-1}\cdot(1-t)^{\beta-1}\mathrm{d}t. \tag{18}$$

Finally, to arrive at the H measure, a normalization is performed to obtain a performance measure bounded by zero and one:

$$\mathrm{H} = 1 - \frac{\int_0^1 Q_C(T(c);b,c)\cdot u(c)\mathrm{d}c}{\pi_0\int_0^{\pi_1} c\cdot u(c)\mathrm{d}c + \pi_1\int_{\pi_1}^1 (1-c)\cdot u(c)\mathrm{d}c}, \tag{19}$$

here $u(c)$ is shorthand notation for $u_{\alpha,\beta}(c)$. The denominator gives the misclassification loss for the worst classifier, namely a random classifier. Note also that the integration over $c \in [0,1]$ corresponds to an integration over $\theta \in [0,+\infty)$, and thus of a ROC curve tangent slope going from plus infinity to zero.

4.3. HOSVM Algorithm for Profit-based Feature Selection and Classification

Since we are dealing with class-imbalanced data, we first redefine the Holdout SVM algorithm to incorporate a resampling step. We propose and empirically study two strategies: random undersampling and a combination of random undersampling with SMOTE, an intelligent resampling technique. The purpose of the algorithm is to find a subset $\mathcal{K}$ ($\mathcal{K} \subseteq \mathcal{F}$) of features, such that the performance of the SVM classifier using this subset's features is maximized, considering a training set $\mathcal{T}$. This set is splitted into a training

subset $\mathcal{TR}$ and a validation subset $\mathcal{V}$. The training subset $\mathcal{TR}$ is resampled into a new subset $\mathcal{TR}'$, where the classifier is constructed, in order to achieve a balanced classification problem. The validation subset is finally used to construct a suitable loss function for the churn prediction problem. Accordingly, the Holdout approach for profit-based feature elimination and classification is provided in Algorithm 1.

Algorithm 1 Holdout algorithm for profit-based feature elimination and classification

Input: The original set of features $\mathcal{F}$
Output: An ordered vector of features $\mathcal{F}^\dagger$

1. $\mathcal{F}^\dagger \leftarrow \emptyset$
2. **repeat**
3. $(\mathcal{TR}, \mathcal{V}) \leftarrow$ Holdout using $\mathcal{T}$
4. $\mathcal{TR}' \leftarrow$ Resampling($\mathcal{TR}$)
5. $\mathbf{\Lambda} \leftarrow$ SVM Training using $\mathcal{TR}'$
6. $\mathcal{I} \leftarrow \text{argmin}_{\mathcal{I}} \sum_{j \in \mathcal{I}} \text{LOSS}^{(-j)}(\mathbf{\Lambda}, \mathcal{TR}', \mathcal{V}),\ \mathcal{I} \subset \mathcal{F}$
7. $\mathcal{F} \leftarrow \mathcal{F} \setminus \mathcal{I}$
8. $\mathcal{F}^\dagger \leftarrow (\mathcal{F}^\dagger, \mathcal{I})$
9. **until** $\mathcal{F} = \emptyset$

The SVM classifier trained in $\mathcal{TR}'$ (Step 5) is given by $\Lambda = (\boldsymbol{\alpha}, b)$, and this information is an input for $\text{LOSS}^{(-j)}(\mathbf{\Lambda}, \mathcal{TR}', \mathcal{V})$, the novel loss functions we propose in this work. Here we suggest to calculate measures MPC, EMPC, and H using subset $\mathcal{V}$ when attribute j is removed. Intuitively, the attribute whose removal leads to a higher profit (or a lower cost, for the H metric) has to be eliminated from the dataset. To adapt these metrics, we first notice that our proposals only differ with the original versions of MPC, EMPC, and H in the computation of the score (and therefore the probability distributions), while the cost and benefits of a given solution as well as the definition of γ are not affected. Following the ideas of the contribution measures for RFE-SVM and HOSVM, we define $s_k^{(-j)}$, i.e. the score of a sample $k \in \mathcal{V}$ when attribute

j is removed as follows:

$$s_k^{(-j)}(\mathbf{\Lambda}, \mathcal{TR}', \mathcal{V}) = \sum_{i \in \mathcal{TR}'} \alpha_i y_i K(\mathbf{x}_i^{(-j)}, \mathbf{x}_k^{v(-j)}) + b \tag{20}$$

where $\mathbf{x}_i^{(-j)}$ is a sample from the resampled training subset when attribute j is removed and $\mathbf{x}_k^{v(-j)}$ means validation object k with feature j removed. To reduce the algorithm's computational complexity, the vector α is assumed to be equal to the solution of Formulation (4) even if one attribute has been removed, as suggested in [20].

The following loss functions are proposed for Step 6 of the algorithm, where the only difference between the original metrics is the redefinition of probability distributions based on the new score formula $s^{(-j)}$:

- **H measure**:

$$\text{LOSS}^{(-j)}(\mathbf{\Lambda}, \mathcal{TR}', \mathcal{V}) = \text{H}(s^{(-j)}) \tag{21}$$

- **Maximum Profit (MPC)**:

$$\text{LOSS}^{(-j)}(\mathbf{\Lambda}, \mathcal{TR}', \mathcal{V}) = \text{MPC}(s^{(-j)}) \tag{22}$$

- **Expected Maximum Profit (EMPC)**:

$$\text{LOSS}^{(-j)}(\mathbf{\Lambda}, \mathcal{TR}', \mathcal{V}) = \text{EMPC}(s^{(-j)}) \tag{23}$$

Using these loss functions in Step 6 of the algorithm leads to the three variants of our proposed approach called HOSVM$_H$, HOSVM$_{MPC}$, and HOSVM$_{EMPC}$, respectively.

Finally, in Step 6 the algorithm determines a set $\mathcal{I}$ of features to be eliminated. While one could choose a single element of $\mathcal{F}$, this would be inefficient if there are

many irrelevant features. On the other hand, removing too many features at a time increases the risk of eliminating relevant features [20].

5. Experimental Results

In this section we report experiments on three churn prediction problems using the proposed model and alternative feature selection approaches. We first describe the datasets we used. Then the experimental setting is presented, followed by the results.

5.1. Description of datasets

The three data sets we used are from class-imbalanced binary-classification problems and will be described next.

- UCI-Telecom: This customer churn dataset available from the UCI repository [2] contains information of 5,000 customers from a telecommunication company, described by 20 attributes.
- Operator 1: This telecommunication dataset was originally studied by [28], and contains data from 47,761 customers described by 47 variables. It was used for benchmarking machine learning methods in [35] under the name of Operator 1 (O1).
- Cell2Cell: This dataset was proposed in [14] as a case study, and was previously used in [35] for benchmarking machine learning methods under the name of D2.

Table 1 summarizes the relevant information for each benchmark dataset:

Dataset	#variables	#examples(min.,maj.)	churn rate
UCI-Telecom	20	(707;4,293)	16.5%
Operator 1	47	(1,761;46,000)	3.8 %
Cell2Cell	73	(406;20,000)	2.0 %

Table 1: Number of variables, number of examples of each class and churn rate for all three datasets.

5.2. Proposed experimental setting

The KDD process [17] [3] is applied to develop all churn prediction models; a well-known methodology which has been successfully used in business analytics, e.g. for churn prediction and credit scoring [6]. The relevant steps of the methodology follow:

- Feature ranking and model selection: The following procedure was used for feature ranking and hyperparameter setting. Training and test subsets are obtained using a 10-fold cross-validation, which is a common procedure for validation in churn prediction [35]. In particular, we used stratified sampling to generate the 10 partitions, making sure that the proportion of churners/non-churners is similar in each fold. Feature ranking and classification is then performed in the training set, and the classification performance is computed by averaging the test results. The training set was resampled considering two alternatives: random undersampling and a combination of random undersampling and SMOTE oversampling. Model selection was performed via grid search along different predefined feature subsets. The following values were studied: $C \in \{2^{-7}, ..., 2^{7}\}$ and $\sigma \in \{2^{-7}, ..., 2^{7}\}$.

- Model stability and influence of the parameters: The performance of all methods was studied for different parameter values to assess their impact on the final classifier.

The toolbox LibSVM [11] was used for standard SVM approaches.

5.3. Results

In this section, a summary of the results is presented to facilitate assessing the best performance of the respective approaches. Tables 2, 3, and 4 summarize the average performance among different feature subsets for each method in datasets UCI-Telecom, Cell2Cell, and Operator 1, respectively.

We consider the following metrics: AUC, EMPC, MPC and H measure. The EMPC and MPC measures are reported in Euro (€) per customer. The best performance among all methods is highlighted in bold type. We also indicate with one asterisk where the performance is significantly worse than the best method at

a 10% significance level, with two asterisks at a 5% significance level, and with three asterisks at a 1% significance level. An independent two-sample t-test is used to make pairwise comparisons between the mean of each approach and the best method for a particular dataset using as hypothesis that the mean performance of the proposed approach is equal to the mean performance of the best available method. Results are displayed for the best resampling strategy, which was random undersampling for each dataset.

	Fisher	RFE	HOSVM$_{EMPC}$	HOSVM$_{H}$	HOSVM$_{MPC}$
AUC	62.4**	63.5*	64.5	64.4	**64.6**
EMPC	2.21**	2.45	2.58	2.55	**2.61**
MPC	2.06**	2.36	2.51	2.49	**2.55**
H	0.064**	0.089	0.092	0.085	**0.094**

Table 2: Average performance for all methods and metrics, UCI-Telecom dataset.

	Fisher	RFE	HOSVM$_{EMPC}$	HOSVM$_{H}$	HOSVM$_{MPC}$
AUC	64.81***	94.09	94.09	**94.13**	**94.13**
EMPC	0.224***	**0.879**	0.860	0.860	0.859
MPC	0.223***	**0.876**	0.859	0.860	0.859
H	0.097***	**0.462**	0.429	0.381	0.385

Table 3: Average performance for all methods and metrics, Cell2Cell dataset.

	Fisher	RFE	HOSVM$_{EMPC}$	HOSVM$_{H}$	HOSVM$_{MPC}$
AUC	49.65**	54.35	54.49	**55.18**	54.47
EMPC	0.006	0.006	**0.008**	0.007	0.007
MPC	0.005	0.006	0.007	**0.007**	0.006
H	0.001***	0.001	**0.002**	0.001	**0.002**

Table 4: Average performance for all methods and metrics, Operator 1 dataset.

From Table 2 (UCI-Telecom dataset) we observe that the proposed method using MPC measure for feature elimination and classifier construction has best

overall performance for all the different metrics. The method outperformed Fisher Score with a 5% significance in all metrics and RFE-SVM with a 10% significance for AUC. While HOSVM with EMPC is never significantly lower than the best method for all metrics, HOSVM with the H measure performed significantly lower with a 5% significance when considering EMPC and MPC as assessment metrics.

From Table 3 (Cell2Cell dataset), the proposed method HOSVM with MPC measure has best AUC, while RFE-SVM performed better for the other metrics. Fisher Score is again outperformed by all other methods with a 1% significance in all metrics, while the other methods never perform significantly lower than the best method.

Finally, from Table 4 (Operator 1 dataset), the proposed method has a better performance for metrics EMPC and H, while HOSVM based on H measure achieves better results in AUC and MPC. Again, Fisher Score is outperformed in AUC (5% significance) and H measure (1% significance), while the remaining approaches are never significantly lower than the best method for all metrics.

In order to analyze the feature selection performance of all methods for different subsets of variables, Figures 2 to 4 summarize the best performance for an increasing number of selected features for all three datasets. For each subset of features, the mean AUC is displayed for the methods Fisher Score, RFE-SVM, and the best proposed method according to this measure (HOSVM based on MPC measure for UCI-Telecom and Cell2Cell datasets, and HOSVM based on H measure for Operator 1 dataset). Results are displayed also for the best resampling strategy in each case.

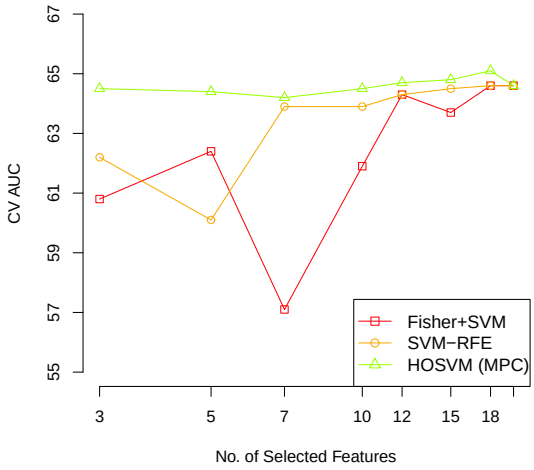

Figure 2: AUC versus the number of ranked variables for different feature selection approaches. UCI-Telecom dataset.

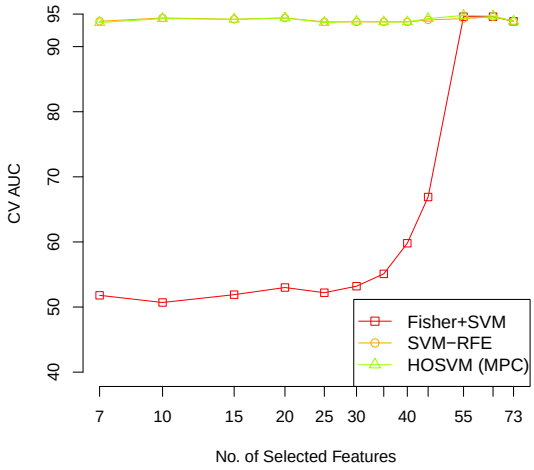

Figure 3: AUC versus the number of ranked variables for different feature selection approaches. Cell2Cell dataset.

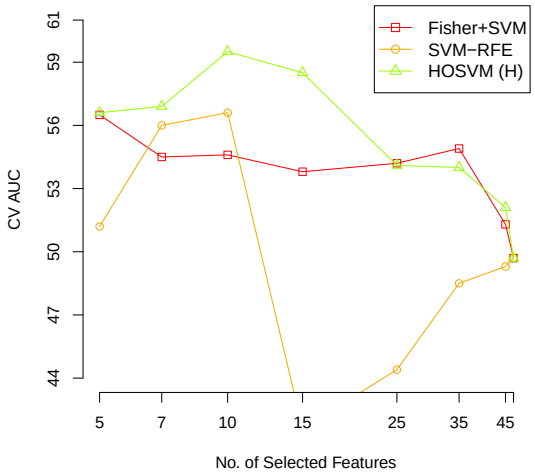

Figure 4: AUC versus the number of ranked variables for different feature selection approaches. Operator 1 dataset.

In Figure 2 (UCI-Telecom dataset) it can be observed that the proposed approach (HOSVM based on MPC measure) achieves the best performance (AUC 0.648 with 18 attributes), and then smoothly decreases its performance. In contrast, Fisher Score and RFE-SVM significantly decrease their performance when removing variables.

For Figure 3 (Cell2Cell dataset), a similar behavior between RFE-SVM and the proposed approach (HOSVM based on MPC measure) is achieved, although best performance is achieved with the latter method (0.948 with 55 attributes). Fisher Score, in contrast, removes relevant attributes after 55 attributes, which significantly decreases the method's performance.

Finally, in Figure 4 (Operator 1 dataset), we observe an important gain by using feature selection compared to the case with all attributes. Best performance is achieved with HOSVM based on H measure (AUC=0.595), while the alternative approaches also help to improve predictive performance but with lower accuracy and in a less stable form.

The previous figures highlight that the proposed approaches outperform al-

ternative feature selection methods, which can be clearly observed since the corresponding line is almost always above the others, i.e., the proposed approaches have higher AUC for the different subset of features. However, no clear trend can be inferred from these figures, which is somehow expected due to the high class-imbalance and overlap that strongly affects the predictive performance.

Three main conclusions can be drawn from the previous results: first, the proposed method clearly performed better than the alternative approaches used in this work, since for all datasets the proposed strategy achieves the maximum performance for a given number of features, and the best overall performance (average performance among all subsets of features), with the only exception being the Cell2Cell dataset, where RFE-SVM has better overall performance when using EMPC, MPC and H measure. Secondly, our experiments demonstrate the usefulness of feature selection in terms of predictive performance, even for low-dimensional applications such as churn prediction, since the best solution was always found when performing an adequate feature selection. Finally, data resampling proved to improve results in all datasets, demonstrating its effectiveness at tackling the class imbalance issue. Undersampling seemed to be more relevant than oversampling in our case, which is somehow expected given the large size of the dataset.

6. Conclusions

In this work we present a backward elimination approach for classification and embedded feature selection using SVM. The proposed method studies three different evaluation measures suitable for class-imbalance problems, and, in particular, for churn prediction problems: the H measure, the MPC metric, and the EMPC measure. While the H measure provides a framework to explicitly consider the misclassification costs as a measure of predictive performance [21], MPC [35] and EMPC [38] go one step further and incorporate the benefits of retention campaigns into the churn prediction task, resulting in very powerful and goal-oriented metrics for model assessment. The main difference between MPC and EMPC is that the latter considers the decision of a potential churner to accept a retention incentive as a random variable, and then computes the expected profit of the retention

campaign for telecommunication companies.

In contrast to the available literature on this topic, which aims at selecting the best model among various classification methods using statistically motivated performance measures, our objective is to provide a framework that allows the adequate selection of the hyperparameters and the right features for classifier construction, focusing on one method, namely Support Vector Machines, but using profit-based performance measures. Our approach presents the following advantages, based on a comparison with other feature selection approaches for Support Vector Machines in churn prediction applications:

- The proposed method allows to explicitly incorporate costs and benefits obtained from the classification task for churn prediction, leading to a feature selection process especially designed for this particular application.
- The proposed approach achieves better predictive performance than other feature selection techniques in churn prediction problems, considering both traditional assessment metrics (such as AUC) and profit-based measures.
- Our strategy is very flexible and allows using different kernel functions for nonlinear feature selection and classification using SVM. Furthermore, it can also be extended to other classification methods, other than SVM.

There are several opportunities for future work. The feature selection process can be extended to other business analytics applications, such as e.g. credit scoring [6, 32]. While the EMPC metric can be adapted to incorporate the costs and benefits of accepting or rejecting loan applicants, logistic regression can be set as the baseline classifier, which is the most common classification method for this task due to regulatory reasons [32]. Additionally, the costs of variable acquisition and usage can be incorporated into the model, enriching the feature selection process. A step in this direction was presented in [25], where the the features' costs are explicitly incorporated into the model via binary variables and a budget constraint. Another venue for future research would be the extraction of business rules from a developed SVM-model in order to gain interpretability; see e.g. [27].

Acknowledgements

Support from the Chilean "Instituto Sistemas Complejos de Ingeniería" (ICM: P-05-004-F, CONICYT: FBO16, www.sistemasdeingenieria.cl) and of the Explorative Scientific Cooperation Programme between KU Leuven and Universidad de Chile 2012-2013 which funded the project entitled "Development of rule-based classification models using profit maximization" (BIL 12/01) is greatly acknowledged. Sebastián Maldonado was supported by FONDECYT projects 11121196 and 1140831. Richard Weber and Alvaro Flores were supported by FONDECYT project 1140831.

References

[1] S. Alelyani, J. Tang, and H. Liu. *Data Clustering: Algorithms and Applications*, chapter Feature Selection for Clustering: A Review. CRC Press, 2013.

[2] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. URL: `http://archive.ics.uci.edu/ml/`.

[3] B. Baesens. *Analytics in a Big Data World*. John Wiley and Sons, 2014.

[4] R.C. Blattberg, B.D. Kim, and S.A. Neslin. *Database Marketing: Analyzing and Managing Customers*. New York: Springer Science+Business Meida, LLC., 2008.

[5] B. Bonev, F. Escolano, and M. Cazorla. Feature selection, mutual information, and the classification of high-dimensional patterns. *Pattern Analysis and Applications*, 11(3-4):309–319, 2008.

[6] C. Bravo, S. Maldonado, and R. Weber. Methodologies for granting and managing loans for micro-entrepreneurs: New developments and practical experiences. *European Journal of Operational Research*, 227(2):358–366, 2013.

[7] D. E. Brown, F. Famili, G. Paas, K. Smith-Miles, L. C. Thomas, R. Weber, R. Baeza-Yates, C. Bravo, G. L'Huillier, and S. Maldonado. Future trends in

business analytics and optimization. *Intelligent Data Analysis*, 15(6):1001–1017, 2011.

[8] J. Burez and D. Van den Poel. Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3):4626–4636, 2009.

[9] K. P. Burnham and D. R. Anderson. *Model Selection and Multimodel Inference*. John Wiley and Sons, 2002.

[10] E. Carrizosa, B. Martín-Barragán, and D. Romero-Morales. Detecting relevant variables and interactions in supervised classification. *European Journal of Operational Research*, 213(1):260–269, 2011.

[11] C.C. Chang and C.J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[12] N. Chawla. *Data mining for imbalanced datasets: An overview*. Springer, Berlin, 2010.

[13] P. Datta, B. Masand, D.R. Mani, and B. Li. Automated cellular modeling and prediction on a large scale. *Artificial Intelligence Review*, 14:485–502, 2000.

[14] Center for Customer Relationship Management Duke University, February 2014. URL: `http://www.fuqua.duke.edu/centers/ccrm`.

[15] M.A.H. Farquad, V. Ravi, and S. Bapi Raju. Churn prediction using comprehensible support vector machine: An analytical crm application. *Applied Soft Computing*, 19:31–40, 2014.

[16] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874, 2006.

[17] U. Fayyad. Data mining and knowledge discovery- making sense out of data. *IEEE Expert-Intelligent Systems and Their Aplications*, 11:20–25, 1996.

[18] J.H. Fleming and J. Asplund. *Human Sigma: Managing The Employee-Customer Encounter*. Gallup Press, New York, 2007.

[19] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh. *Feature extraction, foundations and applications*. Springer, Berlin, 2006.

[20] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

[21] D.J. Hand. Measuring classifier performance: a coherent alternative to the area under the roc curve. *Machine Learning*, 77(1):103–123, 2009.

[22] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *In NIPS*. MIT Press, 2005.

[23] A. Lemmens and C. Croux. Bagging and boosting classification trees to predict churn. *Journal of Marketing Research*, 43(2):276–286, 2006.

[24] H. Li, C.-J. Li, X.-J. Wu, and J. Sun. Statistics-based wrapper for feature selection: An implementation on financial distress identification with support vector machine. *Applied Soft Computing*, 19:57–67, 2014.

[25] S. Maldonado, J. Pï¿½rez, M. Labbï¿½, and R. Weber. Feature selection for support vector machines via mixed integer linear programming. *Information Sciences*, 279:163–175, 2014.

[26] S. Maldonado and R. Weber. A wrapper method for feature selection using support vector machines. *Information Sciences*, 179:2208–2217, 2009.

[27] D. Martens, B. Baesens, T. Van Gestel, and J. Vanthienen. Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research*, 183(3):1446–1476, 2007.

[28] M. Mozer, R. Wolniewicz, D. Grimes, E. Johnson, and H. Kaushansky. Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. *IEEE Transactions on Neural Networks*, 11(3):690–696, 2000.

[29] S.A. Neslin, S. Gupta, W.A. Kamakura, J. Lu, and C.H. Mason. Defection detection: Measuring and understanding the predictive accuracy of customer churn models. *Journal of Marketing Research*, 43(2):204–211, 2006.

[30] J. Reunanen. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3:1371–1382, 2003.

[31] B. Schï¿½lkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, USA., 2002.

[32] L.C. Thomas, J.N. Crook, and D.B. Edelman. *Credit Scoring and its Applications*. SIAM, 2002.

[33] J. Van Hulse, T.M. Khoshgoftaar, A. Napolitano, and R. Wald. Feature selection with high-dimensional imbalanced data. In *Proceedings of the IEEE International Conference on Data Mining Workshops*, pages 507–514, 2009.

[34] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

[35] W. Verbeke, K. Dejaeger, D. Martens, J. Hur, and B. Baesens. New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research*, 218(1):211–229, 2012.

[36] W. Verbeke, D. Martens, and B. Baesens. Social network analysis for customer churn prediction. *Applied Soft Computing*, 14:431–446, 2014.

[37] W. Verbeke, D. Martens, C. Mues, and B. Baesens. Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, 38:2354–2364, 2011.

[38] T. Verbraken, W. Verbeke, and B. Baesens. A novel profit maximizing metric for measuring classification performance of customer churn prediction models. *IEEE Transactions on Knowledge and Data Engineering*, 25(5):961 – 973, 2012.

[39] C.P. Wei and I.T. Chiu. Turning telecommunications call details to churn prediction: a data mining approach. *Expert Systems with Applications*, 23:103–112, 2002.

[40] Z. Zhang and E. Hancock. A hypergraph-based approach to feature selection.

[41] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *ICML '07: Proceedings of the 24th International Conference on Machine Learning*, pages 1151–1157, New York, NY, 2007. ACM.

[42] Z. Zhao, L. Wang, H. Liu, and J. Ye. On similarity preserving feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):619–632, 2013.